HW4

Create website called PasteChart for people to share their data as graphical charts

Use MVA architecture w/namespaces as used in Hw3.

Should work with Composer and classes loaded using Composer’s autoloader

composer.json should list Simpletest ≥ 1.1 as dependency.

Readme.txt file should explain how to set up project as well as how to run simpletest unit tests

Each commit should have less than 150 lines of difference

Break code into issues into a folder “issues”

**Format of issues:**

Number\_short\_title.txt

Ex: 0001\_Set\_up\_folder\_structure.txt

**Contents of issues:**

Title

Owner

Short description of what needed to be done for the issue

**About PasteChart**

* Landing page
  + Title and h1 tag – “PasteChart”
  + “Share your data in charts!”
  + text field with label=”Chart Title”
  + large textarea with placeholder saying allowed format for data
  + Share button
    - Check if data conforms to format (done in both client and also server side bc client shouldn’t trust client)
      * If not display error on page, don’t submit form, and landing page with cleaned version of the data should be drawn
      * If it does conform: data should be md5 hashes using PHP hash function and tuple (md5, title, data) should be stored into db
      * Script should draw page with title and h1 heading: “XXXXX LineGraph – PasteChart” where XXXXX is the md5 hash
      * Below h1 heading: line graph with title as given by user and using user provided data
      * \*Below that, print the lines (URL of this page should be same as LineGraph URL below):
        + Share your chart and data at the URLs below:
        + As a LineGraph: BASE\_URL/?c=chart&a=show&arg1=LineGraph&arg2=XXXXX
        + As a PointGraph: BASE\_URL/?c=chart&a=show&arg1=PointGraph&arg2=XXXXX
        + As a Histogram: BASE\_URL/?c=chart&a=show&arg1=Histogram&arg2=XXXXX
        + As XML data: BASE\_URL/?c=chart&a=show&arg1=xml&arg2=XXXXX
        + As JSON data: BASE\_URL/?c=chart&a=show&arg1=json&arg2=XXXXX
        + As JSONP data: BASE\_URL/?c=chart&a=show&arg1=jsonp&arg2=XXXXX&arg3=javascript\_callback
      * Other urls should work and produce a page similar to the one specified with title and h1 = arg1 and actual displayed graph or data being of type arg1.
      * For XML one, create your own XML language that you define in a file called **chart.dtd** included in you Hw4 folder
      * For JSON one, let YYYYY = JSON data output. If arg1=json, let arg3=foo in corresponding jsonp url
        + JSONP response should look like: foo(YYYYY);
* Define a constant BASE\_URL in your config.php script as url of your website
* Data format
  + Comma separated values representing points to be plotted
    - 1 per line
    - 50 lines of at most 80 characters
    - 1st coordinate – text label, cannot be empty string
      * other coordinates can be empty string if values are missing
        + ex: Aug,,10.1
    - remaining coordinates represent values corresponding to that text label from up to 5 sources
      * ex: Month of year, rabbit pop (in thousands), wolf population (in thousands)
        + Jan,600,5.4
        + Feb,450,5.0
* Write at least a few Simpletest unit tests to test that the server side checking is valid
* Enhance the **chart.js** code to handle k-tuples for points rather than just 2-tuples for points